**CUSTOMER CHURN ANALYSIS**

**DEFINITION**

Customer churn or customer attrition is the phenomenon when a company’s customers stop using their product or service during a certain time frame and end their association. A high churn means that a higher number of customers no longer want to purchase goods and services from the business. Businesses are very keen on measuring churn because keeping an existing customer is far less expensive than acquiring a new customer. Customer churn can prove to be a roadblock for an exponentially growing organization and a retention strategy should be decided in order to avoid an increase in customer churn rates.

**HOW TO AVOID HIGH CHURN RATES**

New business use marketing and sales budget to gain additional customers. Big businesses have existing customers and they will often have a higher volume of product/service consumption and can generate additional customers through referrals.

A customer can be retained by providing good product and customer service to them. But still many companies have high customer churn rates. So, the most effective way for a company to prevent attrition of customers is by knowing that what customer truly wants. The vast volumes of data collected about customers can be used to build customer churn prediction models so, company can know how many customers are most likely to defect and company can prioritise focused marketing efforts on that subset of their customer base.

**OUR DATASET**

We are provided customer data from IBM Sample Data Sets which consists customer’s information like their ID, gender, internet service charges, tenure, etc in 21 columns and 7043 rows.

These columns are : -

customerID, gender, SeniorCitizen, Partner, Dependents, tenure, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges, Churn.

**PROBLEM DEFINITION**

We have to examine the dataset and see if the attrition is present or not. We have to create different machine learning customer churn prediction models and choose the best performing model.

**DATA ANALYSIS**

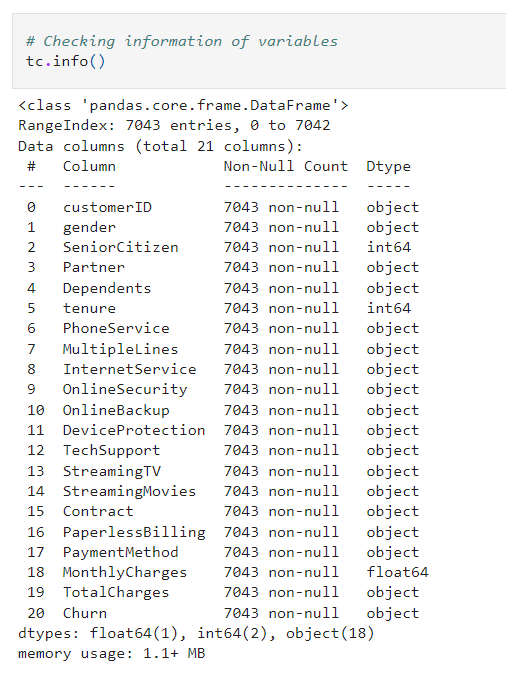
First we’ll be importing dataset “telecom\_customer\_churn.csv” with the help of pandas library



Putting Dataset in DataFrame

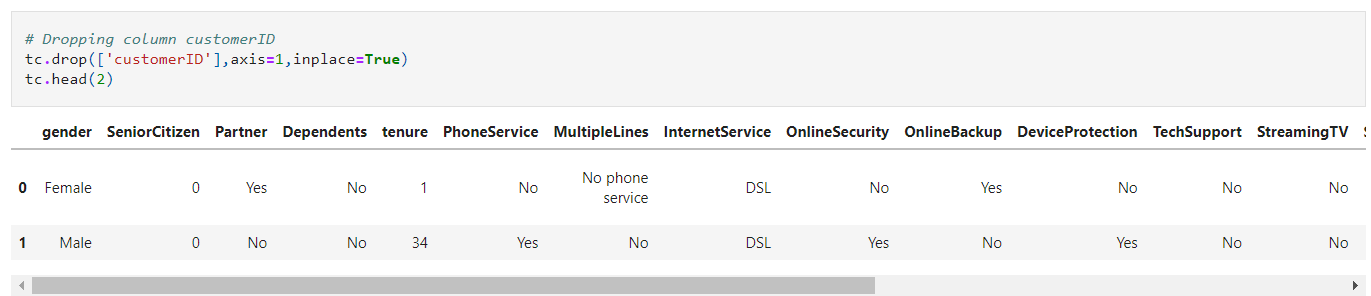


Checking information of columns



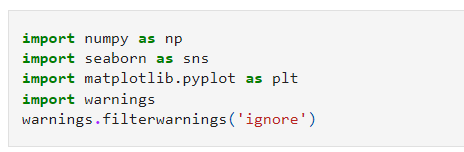
\* We haven’t found any null values in the dataset

Dropping column customerID

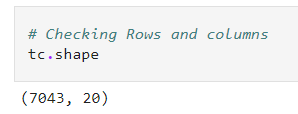


**EXPLORATORY DATA ANALYSIS**

Importing important libraries



Shape of Dataset

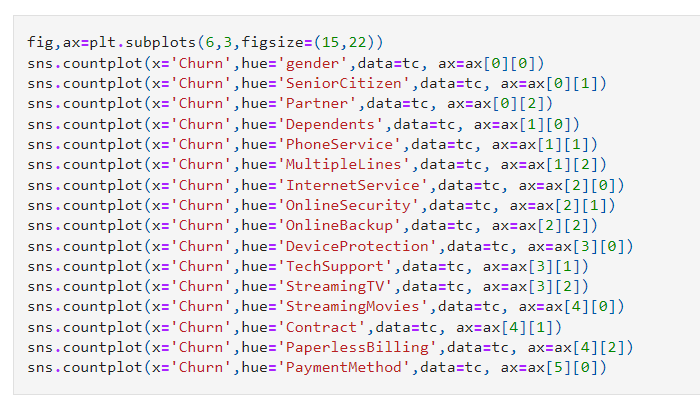


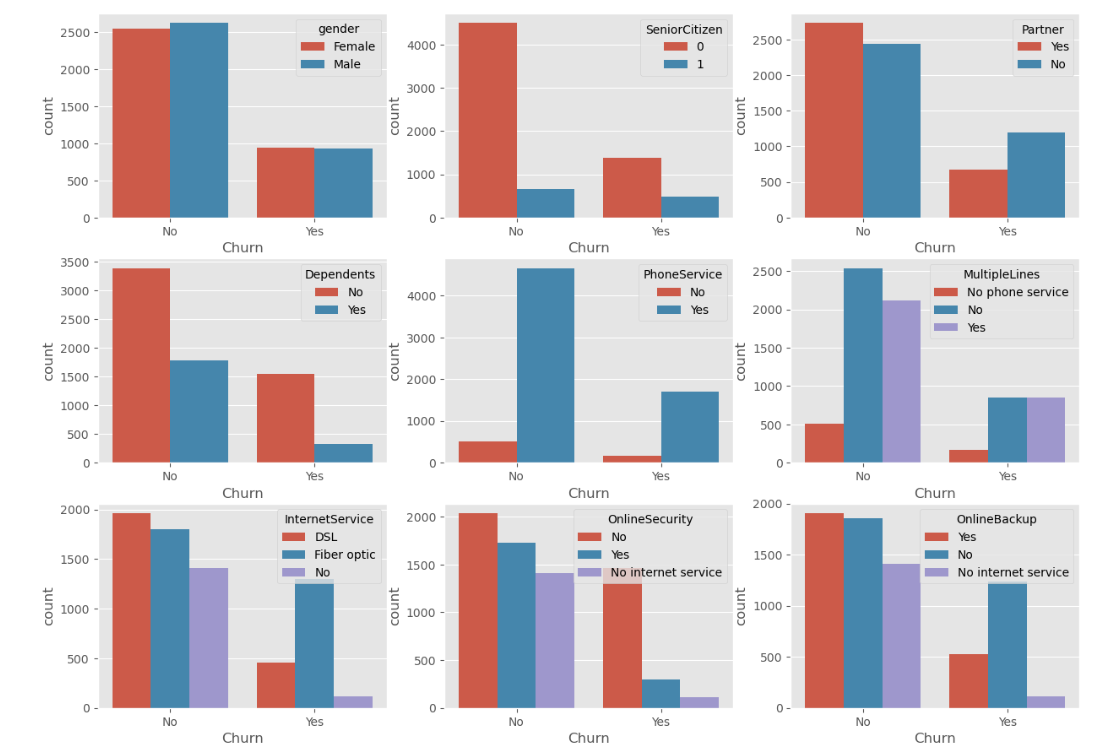
**PLOTS USED : -**

**COUNTPLOT**

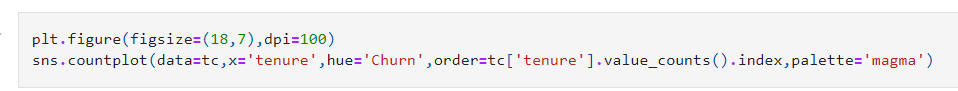
1. Countplot is a method which is used to show the counts of observations in each categorical bin using bars.

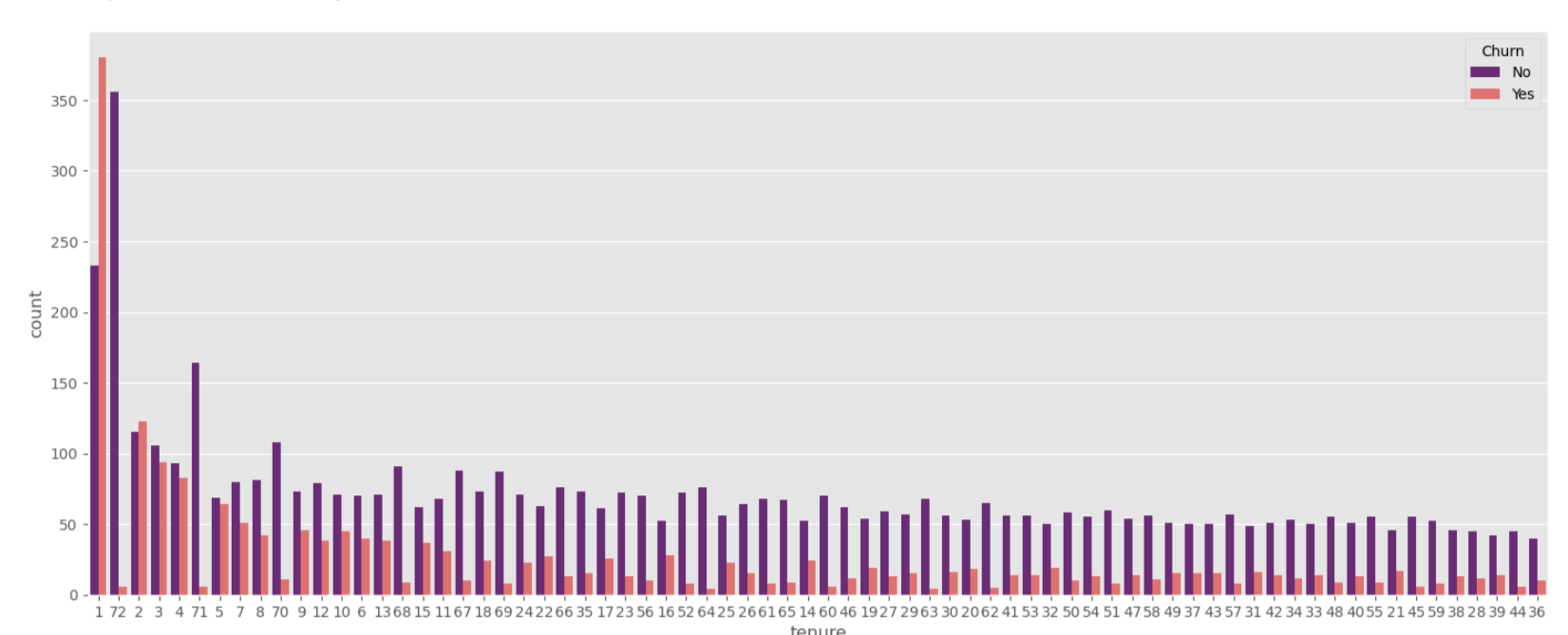
We’ve used countplot from seaborn to count the number of people who are going to defect with respect to the different variables. We’ve given variable ‘Churn’ on x-axis and other variables in hue.



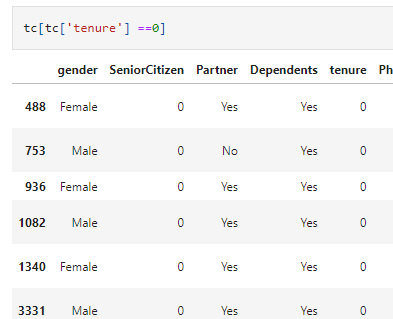


1. We’ll count how many people of different age have yes or no churn. Here we will use countplot, where x is ‘tenure’ with hue ‘Churn’ and we will use ‘magma’ in palette.





* Checking how many rows of variable ‘tenure’ is containing 0

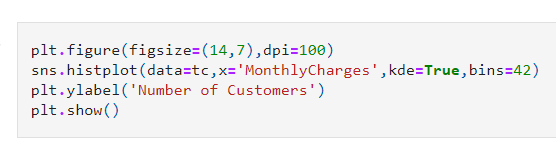


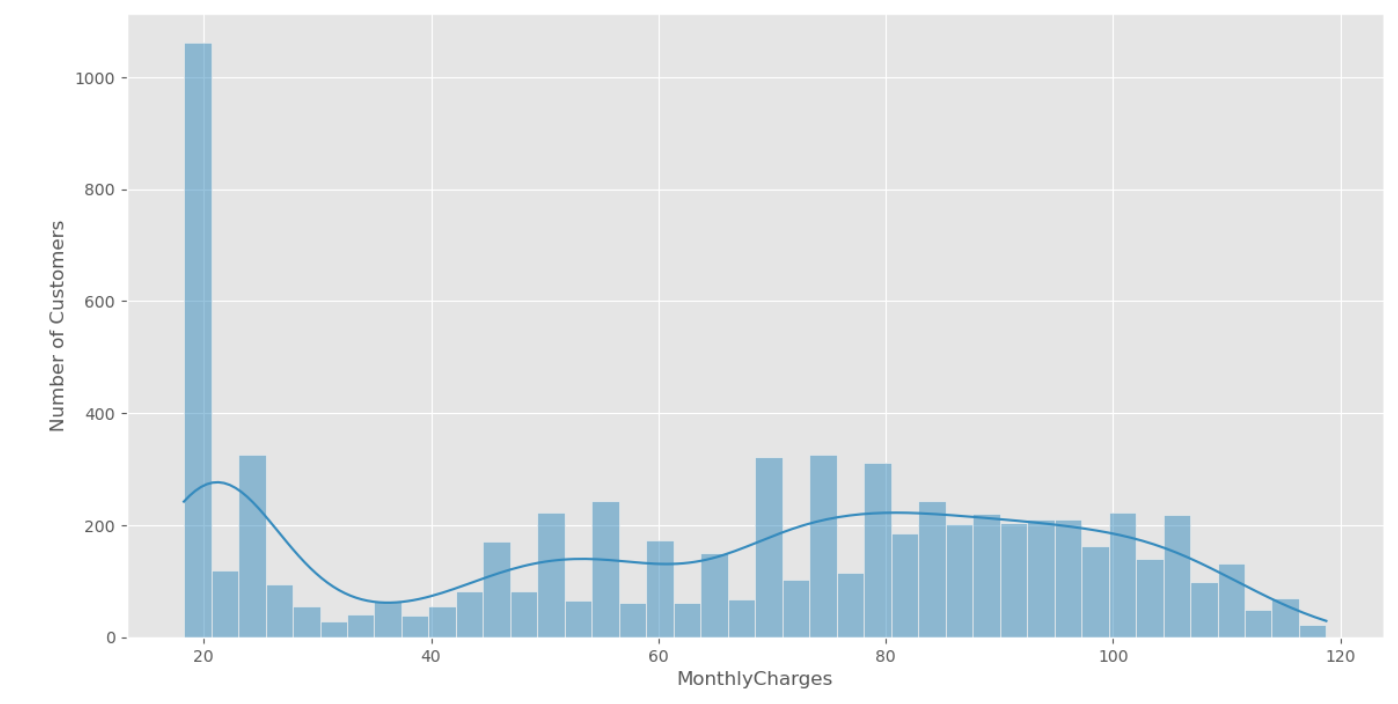
**In the above table tenure is 0 and there is no total charges so we'll drop these rows**

**HISTPLOT**

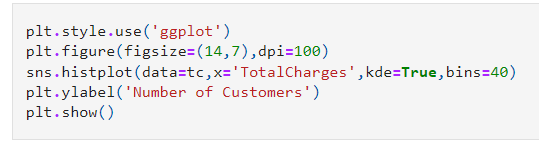
A histogram is a classic visualization tool that represents the distribution of one or more variables by counting the number of observations that fall within discrete bins.

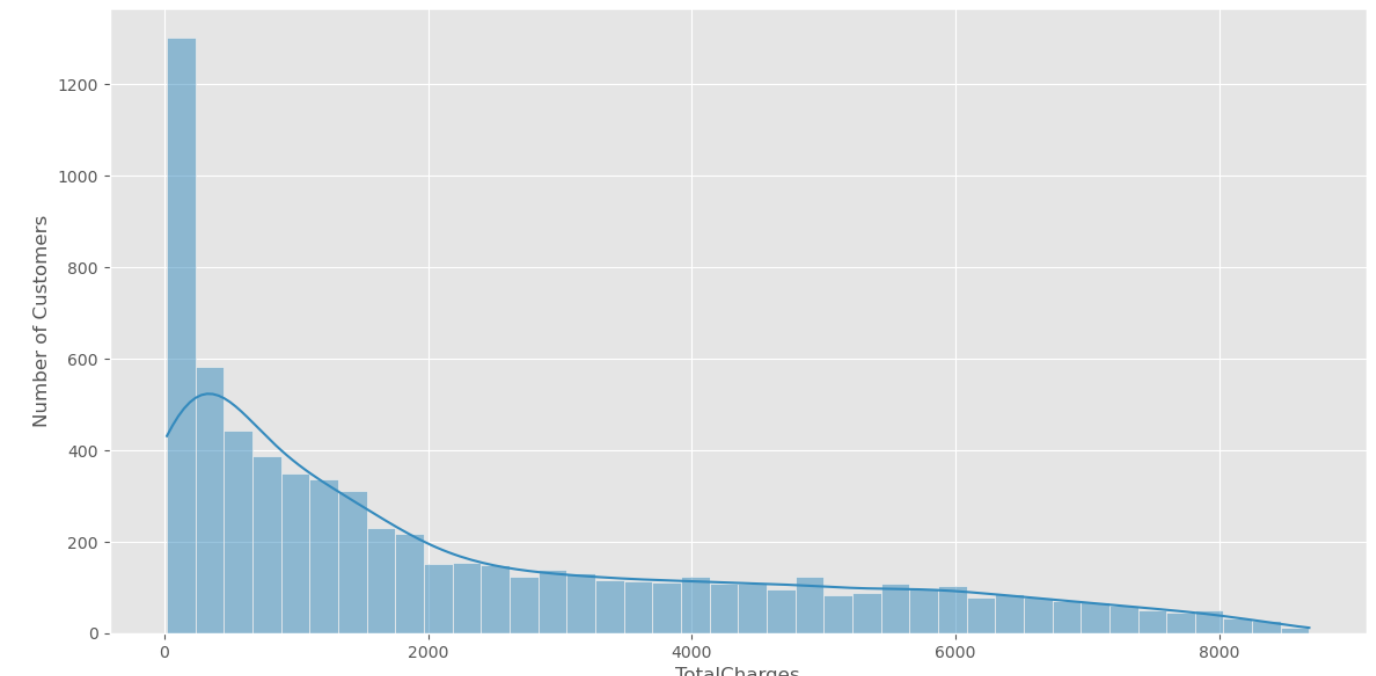
1. We’re using histplot from seaborn to represent distribution of number of customers by variable ‘MonthlyCharges’. Here we’ve given variable ‘MonthlyCharges’ to x-axis and we have given kde = true which means the line showing in the plot.





1. Converting variable ‘TotalCharges’ into float by changing ‘astype(float)’, and using histplot to show the distribution of number of customers by ‘TotalCharges’. Here x-axis have TotalCharges with kde=True



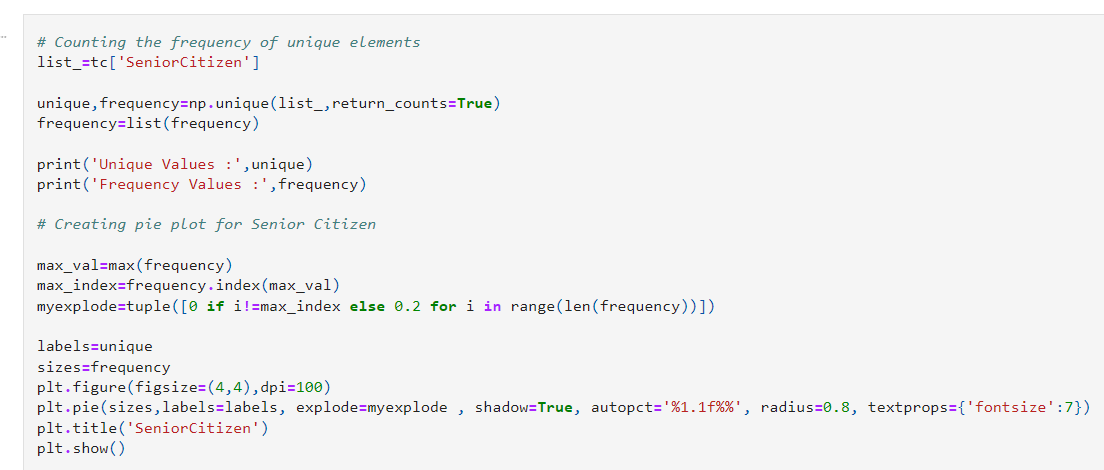


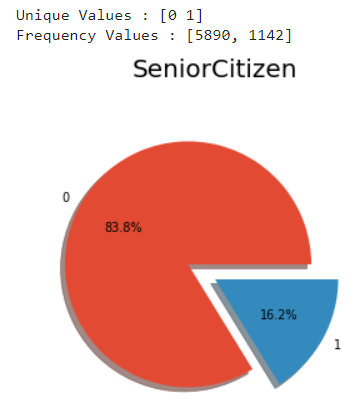
**PIEPLOT**

Step – 1 Creating list for variable to return the counts of unique values and their frequency

Step – 2 Selecting maximum value from frequency and creating variable ‘myexplode’,we will use ‘for’ loop for getting maximum value and explode it by 0.2 from the pie and other pieces will remain intact to the pie.

Step – 3 Creating pie plot, firstly we’ll create variables for parameters like labels and sizes. We’ll make the shadow true and use autopct, it **enables you to display the percent value using Python string formatting,** autopct='%1.1f%%' means that **for each pie wedge, the format string is '1.1f%'.**

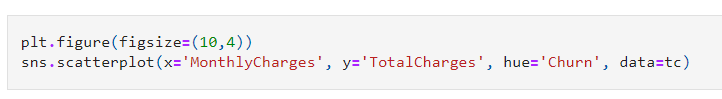


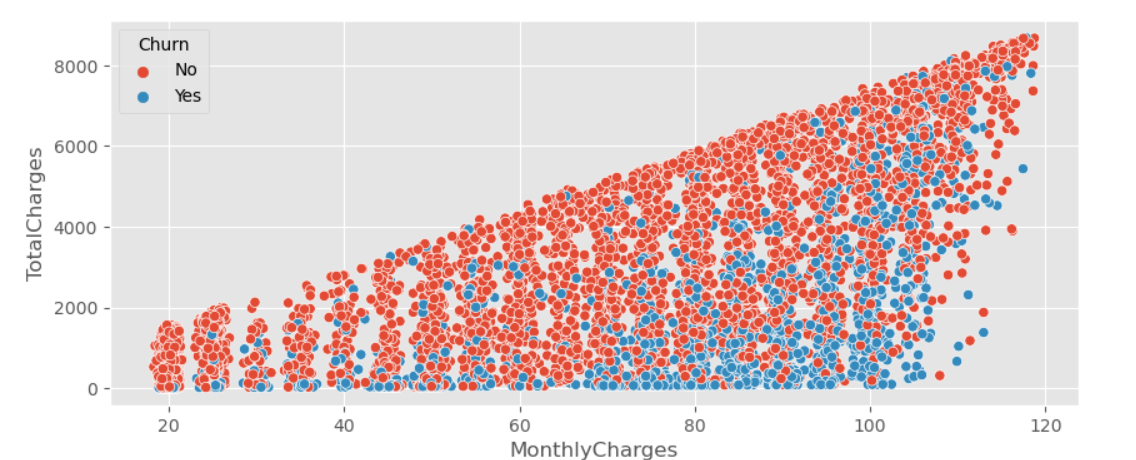


**SCATTERPLOT**

Scatter plots are the **graphs that present the relationship between two variables in a data-set**.

**We’ll see the relationship between ‘MonthlyCharges’ and ‘TotalCharges’ by using scatter plot from seaborn, x-axis – ‘MonthlyCharges’, y-axis – ‘TotalCharges’, hue – ‘Churn’**

****

****

**PRE - PROCESSING**

**CONVERTING CATEGORICAL DATA INTO INTEGER**

**Some of the variables are in categorical form and python has some methods which don’t work on categorical data so we have converted them in integer.**

**LABELENCODER**

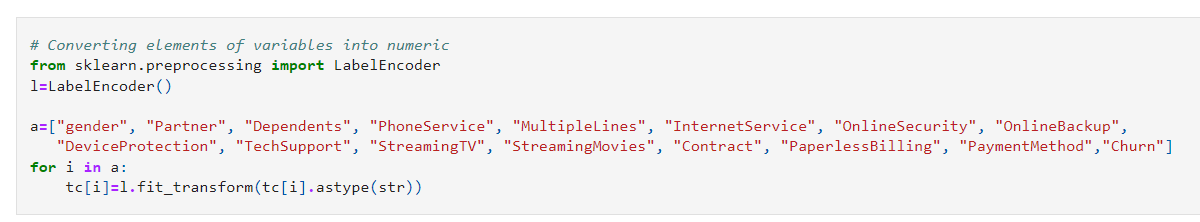
LabelEncoder will convert the categorical elements into numbers like 0, 1, 2, etc.

Step – 1 Import LabelEncoder from library sklearn.preprocessing

Step – 2 Create variable for LabelEncoder

Step – 3 Create list for categorical variables

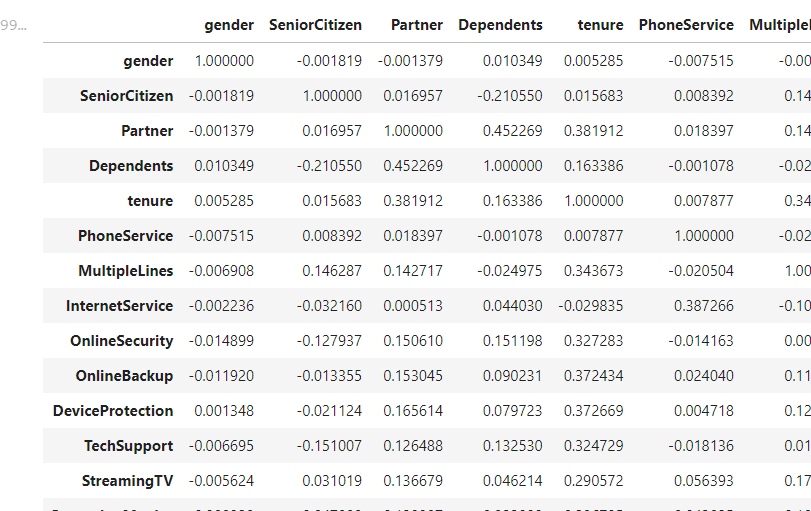
Step – 4 Create a for loop and encode all the categorical variables



Now all the variables are integers

Checking Correlation among the variables

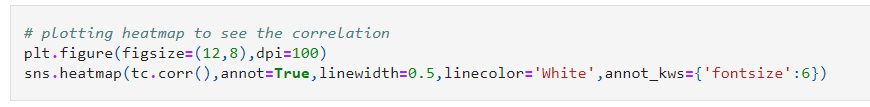


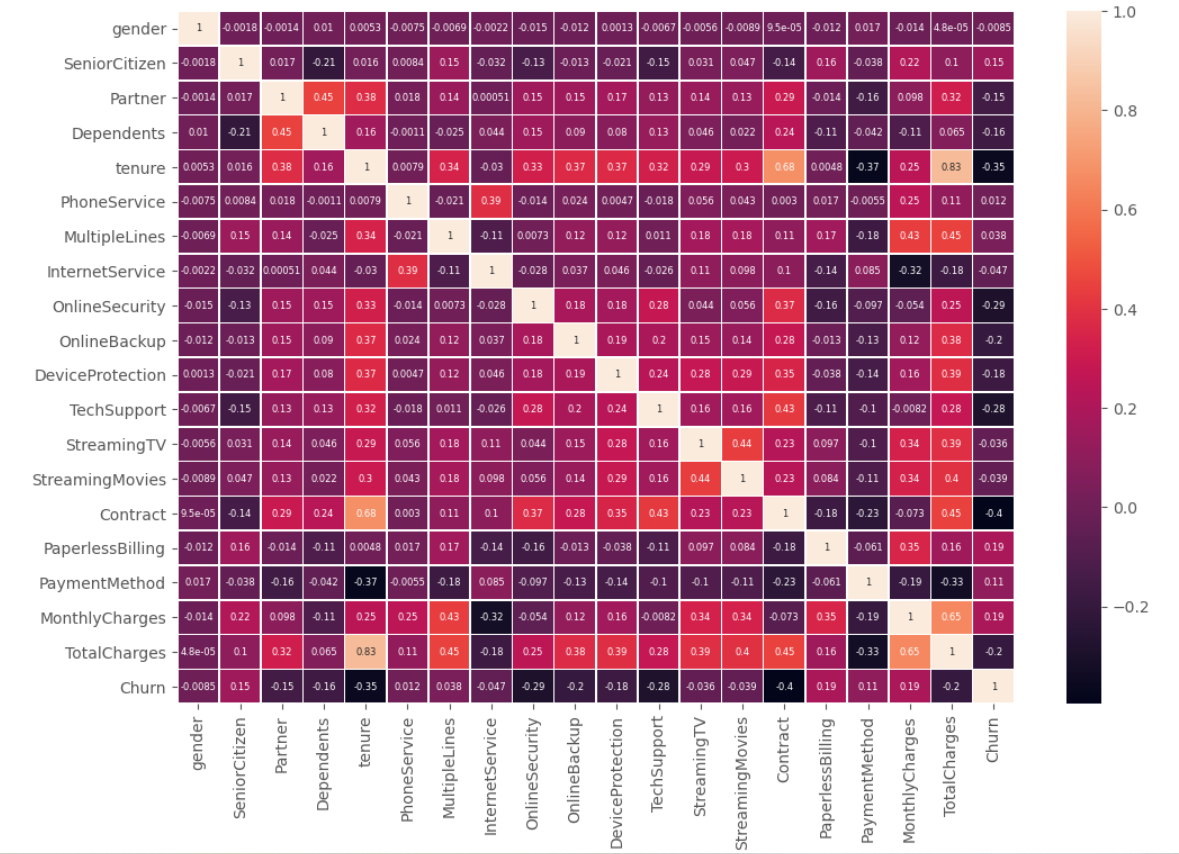


Visualization of Correlation

We’ll use heatmap,  **A heatmap contains values representing various shades of the same colour for each value to be plotted**. Usually the darker shades of the chart represent higher values than the lighter shade.

We’ll make the annotations true, width between the lines is 0.5 and color of lines are white.





**Key Observations : -**

**Most variables are correlated with each other but not with Churn**

**tenure is highly correlated with contract and Total Charges**

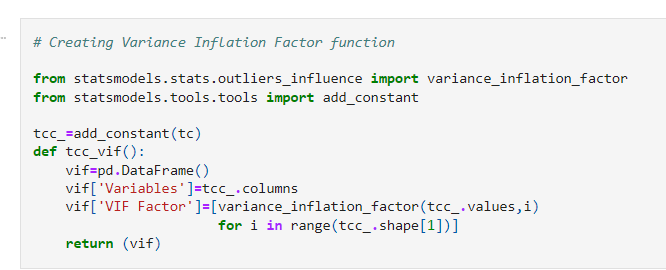
**VARIANCE INFLATION FACTOR (VIF)**

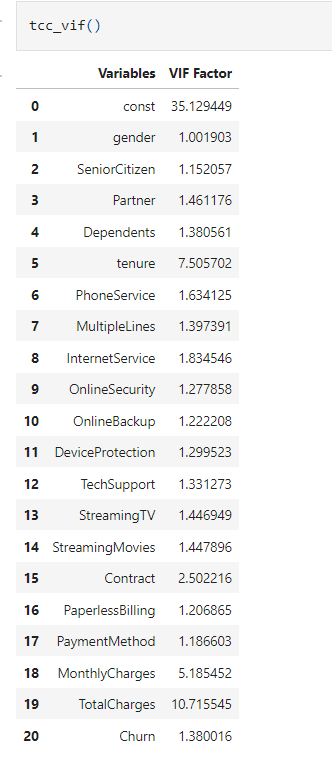
Step – 1 Importing variance\_inflation\_factor from statsmodels.stats.outliers\_influence, Variance inflation factor (VIF) is a measure of the amount of multicollinearity in a set of multiple regression variables.

Step – 2 Import add\_constant from statsmodels.tools.tools, it adds a constant term to the linear equation it is fitting

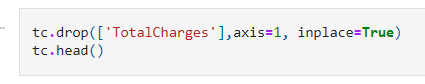
Step – 3 Creating variable to store add\_constant

Step – 4 Creating function named tcc\_vif() for calculating variance inflation factor of all the integer variables and making ‘for’ loop for giving the shape.

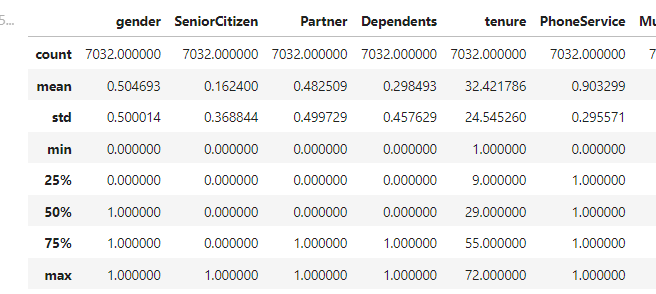




**Tenure and Total Charges have highest variance inflation factor, we'll drop Total charges as it has the highest inflation factor**



**Checking Statistics of variables like mean, median, standard deviation, etc**

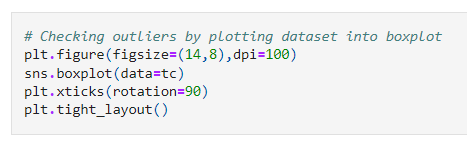


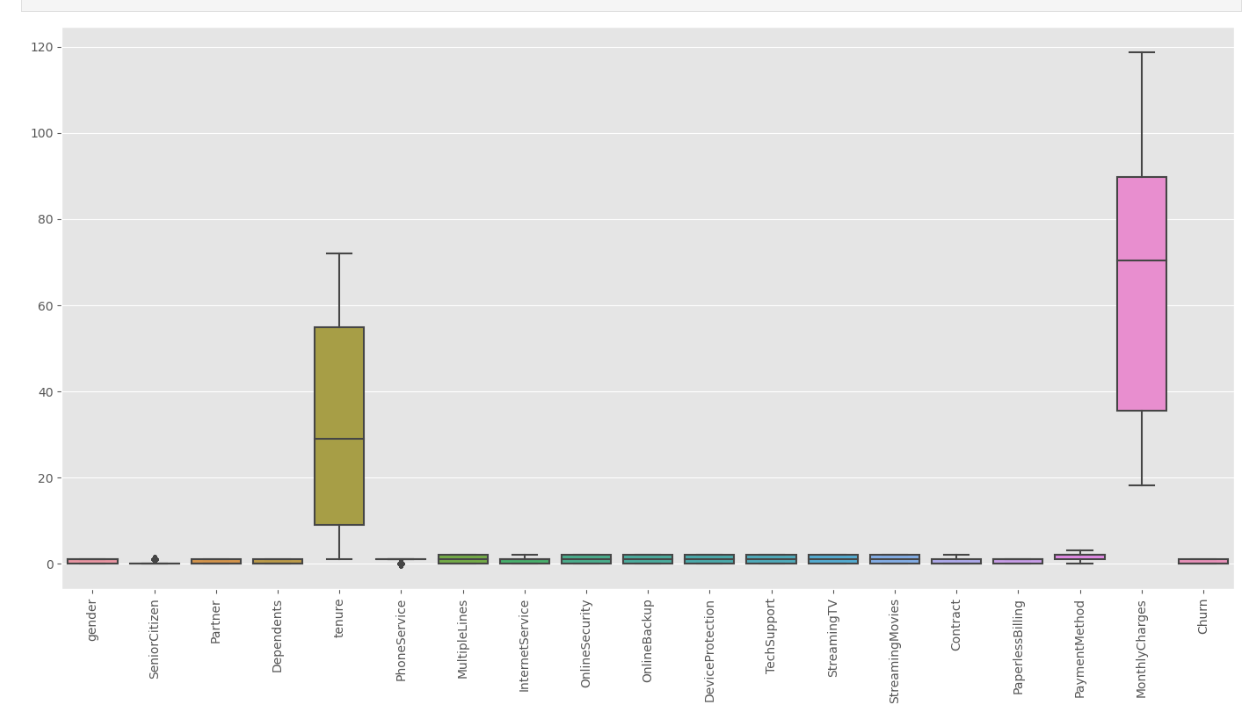
**BOXPLOT**

A **Box Plot** is also known as **Whisker plot**is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum. In the box plot, a box is created from the first quartile to the third quartile, a vertical line is also there which goes through the box at the median. Here x-axis denotes the data to be plotted while the y-axis shows the frequency distribution.

Checking Outliers

An Outlier is a data-item/object that deviates significantly from the rest of the (so-called normal)objects. They can be caused by measurement or execution errors.





**Phone service and senior citizen has some outliers so we will drop these two columns**



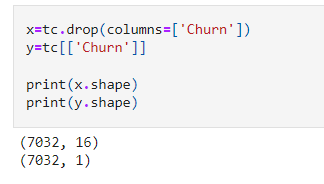
**No outliers are present now**

**MACHINE LEARNING MODELS**

A machine learning model is a file that has been trained to recognize certain types of patterns. You train a model over a set of data, providing it an algorithm that it can use to reason over and learn from those data.

We will create variables for the input variables and target variable

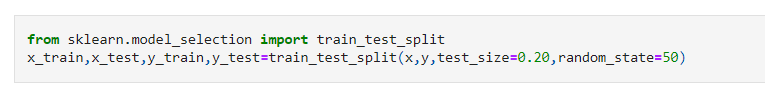
‘x’ Is for input variables so we’ll drop the column ‘Churn’ for ‘x’ & ‘y’ variable is for our target variable which is churn, so we’ll only include ‘churn’ in ‘y’.



‘x’ has 7032 rows and 16 columns while ‘y’ also has 7032 rows but only 1 column.

Now we will train our data so it can learn the independent variables and able to predict pattern of the output variable.

Importing train\_test\_split from sklearn.model\_selection library. Giving the variable ‘x’ and ‘y’ to the train\_test\_split. We’re going to set test size 20 % and training 80%.



**STANDARD SCALER**

Python sklearn library offers us with StandardScaler() function to standardize the data values into a standard format.

**CONCLUSION**

**PRE – PROCESSING**

**MACHINE LEARNING MODELS**

**CONCLUSION**